

---

· Szerb Köztársaság ·  
· Belgrádi Műszaki és Művészeti Főiskolai Tanulmányok Akadémiája ·  
· Villamosmérnöki és Számítástechnikai Főiskola Tanszék ·  
· 2026 ·



---

## TIZEDIK JUBILEUMI ÁLLAMI PROGRAMOZÁSI VERSENY

---

### FELADATOK SZÖVEGE

MAGYAR NYELVEN

Három feladat · 180 perc · 150 pont összesen

### A VERSENY SZABÁLYAI

- Minden feladat maximálisan **50 pontot** ér.
- Minden feladat elején be kell vinni a szükséges bemeneti értékeket.
- Minden feladatnak **5 kimenete** van, amelyeket a feladat elején megadott bemeneti értékek alapján kell kiszámítani.
- Minden kimenet **10 pontot** ér ha helyes, **0-t** ha helytelen.
- A versenyzőknek **EGYÉRTELMŰEN** kell jelölniük minden kimenetet a részfeladat jelzőjével (pl. **1.1:**, **1.2:** stb.). Ha a kimenet nincs egyértelműen jelölve, a versenyző 0 pontot kap azért a kimenetért. Ha a program hibával leáll bármely pillanatban, a versenyzőnek az addig a képernyőn látható kimenetek kerülnek értékelésre. Azonos pontszám esetén az a versenyző kap jobb helyezést, aki korábban adta be a feladatot.
- A verseny **180 percig** tart.

## 1. FELADAT

Adjon meg két ötjegyű számot **a** és **b** (szóközzel elválasztva, egy sorban). Feltételezzük, hogy a számok helyesen vannak megadva – nem szükséges ellenőrizni, hogy a megadott számok valóban ötjegyűek-e. Minden részfeladatnak pontosan **egy kimeneti sora** van – egyértelműen jelöljük meg nyomtatás előtt (pl. **1.1:...**).

**A** Összegek, szorzatok és különbség

Írjon ki egy sorban öt számot: az **a** szám számjegyeinek összegét, a **b** szám számjegyeinek összegét, az **a** szám számjegyeinek szorzatát, a **b** szám számjegyeinek szorzatát, és az **a** szám számjegyösszege és a **b** szám számjegyösszege különbségének abszolút értékét (ebben a sorrendben, szóközzel elválasztva).

**B** Új **c** szám képzése

Képezzen egy új ötjegyű **c** számot úgy, hogy az  $i$  pozícióban lévő minden egyes számjegy ( $i = 0$  az egyesek számjegyéhez,  $i = 1$  a tízesek,  $i = 2$  a százaskok,  $i = 3$  az ezresek,  $i = 4$  a tízezresek számjegyéhez) a következő képlettel számítható:

$$c[i] = (a[i] + b[i] + i) \bmod 10$$

Írja ki csak a **c** számot.

**C** Számjegyek előfordulásainak tömbje

Képezzen egy 10 elemű **d** tömböt úgy, hogy **d[i]** az  $i$  számjegy összes előfordulásainak számát képviselje az **a**, **b** és **c** számokban együttesen. Írja ki a **d** tömb elemeit egy sorban, szóközzel elválasztva (pontosan 10 szám, **d[0]**-tól **d[9]**-ig).

**D** A **mat** mátrix – középső elem

Képezzen egy  $10 \times 10$ -es **mat** mátrixot a következő szabályok szerint:

- Az első sor (0. sor) a **d** tömb.
- Minden következő  $i$  sor ( $1 \leq i \leq 9$ ) az  $i-1$  sorból kapható úgy, hogy az előző sor összes elemét egy hellyel **balra** ciklikusan toljuk, majd minden elemhez hozzáadjuk az  $i$  értékét, csak az eredmény egységes számjegyét megtartva (azaz  $n \bmod 10$ ).

Írja ki csak a középső elem értékét: **mat[5][5]**.

**E** Mátrix forgatása – **rot[2][8]** elem

Képezzen **rot** mátrixot úgy, hogy a **mat** mátrixot  $90^\circ$ -kal **óramutató járásával megegyező irányba** forgatja. Írja ki csak a **rot[2][8]** elem értékét.

## 2. FELADAT

Adjon meg egy **s1** karakterláncot **3·N** ( $3 \leq N \leq 12$ ) karakter hosszan, amely az angol ábécé kis- és nagybetűiből és számjegyekből áll. Feltételezzük, hogy a karakterlánc helyesen van megadva. **N**-t nem kell külön megadni – a karakterlánc hosszából kell kiszámítani. A 0 index a karakterlánc első karakterének felel meg (pl. **s1[0]** az első karakter). Minden részfeladatnak pontosan **egy kimeneti sora** van – egyértelműen jelöljük meg nyomtatás előtt (pl. **2.1:...**).

**A** Alapstatisztika és rendezettség

Írjon ki egy sorban négy értéket (szóközzel elválasztva): a nagybetűk számát, a kisbetűk számát, a számjegyek számát, és **DA** szót ha a karakterlánc **rendezett**, egyébként **NE**. Rendezettnak tekintendő az a karakterlánc, amelyben nem létezik **két egymást követő azonos típusú karakter** (típusok: kisbetű, nagybetű, számjegy).

**B** Három sorozat – két karakter

Az **s1** karakterláncból képezzen 3 sorozatot, mindegyik **N** hosszú, úgy hogy:

- **sek1** a 0, 3, 6, 9, ... pozíciókban lévő karaktereket tartalmazza (3-mal való osztás maradéka 0),
- **sek2** az 1, 4, 7, 10, ... pozíciókban lévő karaktereket tartalmazza (maradéka 1),
- **sek3** a 2, 5, 8, 11, ... pozíciókban lévő karaktereket tartalmazza (maradéka 2).

Írjon ki egy sorban **pontosan két karaktert**, szóközzel elválasztva: az első sorozat utolsó karakterét és a harmadik sorozat első karakterét. Formátum: **sek1[N-1] sek3[0]**.

**C** Karakterátalakítás – két karakter

Minden sorozatban végezze el a következő karakterátalakítást:

- **Kisbetűket** ciklikusan **3 hellyel előre** tolja az angol ábécében ('a' → 'd', 'b' → 'e', ..., 'w' → 'z', 'x' → 'a', 'y' → 'b', 'z' → 'c').
- **Nagybetűket** a megfelelő kisbetűkre cseréli ('A' → 'a', 'B' → 'b', ..., 'Z' → 'z').
- **Számjegyeket** a következő számjegyre cseréli ciklikusan ('0' → '1', '1' → '2', ..., '8' → '9', '9' → '0').

Az átalakítás minden sorozat minden karakterére függetlenül vonatkozik. A **sek1**-ből kapott átalakított sorozatot **t1**-nek, a **sek2**-ből **t2**-nek, a **sek3**-ból **t3**-nak nevezzük. Írjon ki egy sorban **pontosan két karaktert**, szóközzel elválasztva: **t1[N-1] t3[0]**.

**D** Összefűzés és karaktergyakoriságok

Fűzze össze az átalakított sorozatokat (sorban: első, második, harmadik) egy **s4** karakterlánccá. Írjon ki egy sorban **négy értéket**, szóközzel elválasztva: az **s4**-ben **legtöbbször** előforduló karaktert, azt a számot, a **legkevesebbszer** előforduló karaktert, és azt a számot. Csak az **s4**-ben legalább egyszer előforduló karaktereket veszi figyelembe. Ha több ilyen karakter van, azt kell választani, amelynek **kisebb az ASCII értéke**. Formátum: **max\_kar max\_broj min\_kar min\_broj**.

**E****Négyzetes mátrix és forgatás – `mat[0][K-1]` és `rot[K-1][0]` elemek**

Határozza meg  $K = \text{ceil}(\text{sqrt}(\text{len}(s4)))$  értékét – a legkisebb négyzetes `mat` mátrix méretét, amelybe `s4` soronként beírható (ahol `len(s4)` az `s4` karakterlánc hossza, a `ceil` és `sqrt` a felfelé kerekítés és a négyzetgyök függvények). Az üres helyeket `'@'` karakterrel töltsse fel. Majd képezzen `rot` mátrixot úgy, hogy a `mat` mátrixot  $90^\circ$ -kal **balra** (az óramutató járásával ellentétes irányba) forgatja, a `rot[i][j] = mat[j][K-1-i]` képlet szerint. Írjon ki egy sorban **három értéket**, szóközzel elválasztva:  $K$  értékét, majd a `mat[0][K-1]` elemet, aztán a `rot[K-1][0]` elemet. Formátum:  $K$  `mat[0][K-1]` `rot[K-1][0]`.

## 3. FELADAT

Pingo pingvin egy jeges területen találta magát, amelyet Leo, a régi tengeri oroszlán irányít. A területet  $10 \times 10$ -es mezőmátrix ábrázolja. A koordinátarendszer origója a mátrix bal felső sarkában van  $((0,0)$  mező) – **a sorok és oszlopok 0-tól 9-ig indexeltek.**

Adjon meg négy egyjegyű számot **r1 c1 r2 c2** (szóközzel elválasztva, egy sorban). Ezek két pozíciót határoznak meg:

- **(r1, c1)** – **Pingo** (a pingvin) kezdeti pozíciója,
- **(r2, c2)** – **Leo** (a tengeri oroszlán) kezdeti pozíciója.

Minden részfeladatnak pontosan **egy kimeneti sora** van. Csak a **3.5-ös** részfeladat rajzolja ki a mátrixot.

## Mi a Manhattan-távolság?

Két  $(x_1, y_1)$  és  $(x_2, y_2)$  pontra egy mátrixhálóban a **Manhattan-távolság** a koordinátakülönbségek abszolút értékeinek összege:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Ez az a **lépésszám**, amely szükséges ahhoz, hogy az első pontból a másodikba eljussunk, ha csak vízszintesen vagy függőlegesen mozoghatunk (sohasem átlósan) – képzelje el a mozgást egy téglalap alapú utcahálózatú városban, innen ered a neve (New York városának Manhattan szigetéről).

**Példa:** a  $(3, 3)$  és  $(7, 7)$  pontok közötti Manhattan-távolság  $|3 - 7| + |3 - 7| = 4 + 4 = 8$  lépés (nem  $\sqrt{32} \approx 5,66$ , ahogy az euklideszi távolság esetén lenne).

## A játék szabályai (egyszerűsített):

- Pingo a mátrix **legközelebbi széle felé** halad (Manhattan-távolság szerint minden oldaltól). Ha több szélről egyforma távolságra van, prioritás szerint választja az irányt: **FEL** (a 0. sor felé), **JOBBRA** (a 9. oszlop felé), **LE** (a 9. sor felé), **BALRA** (a 0. oszlop felé). Egyenes vonalban halad, soronként egy mezőt. A program **GORE/DESNO/DOLE/LEVO** értékeket ír ki.
- Pingo akkor számít **megszököttnek**, ha elér bármely szélső sort vagy oszlopot.
- Leo nem mozog ebben az egyszerűsített modellben. A kimenetelt a távolságok összehasonlítása határozza meg:
  - ha **Pingo legközelebbi szélről mért távolsága**  $\leq$  **Pingo és Leo közötti Manhattan-távolság**, Pingo megszökik,
  - ellenkező esetben Leo elkapja Pingót a kezdeti **(r1, c1)** pozíciójában.

## A Manhattan-távolság

Írja ki Pingo és Leo kezdeti pozíciói közötti Manhattan-távolságot.

## B Pingo távolsága a legközelebbi szélről

Írja ki Pingo legkisebb Manhattan-távolságát a mátrix bármely szélétől  $(\min(r1, 9-r1, c1, 9-c1))$ .

## C Pingo mozgásának iránya

Írja ki azt a szót, amely megjelöli az irányt, amelybe Pingo mozogni fog, prioritás szerint: **GORE**, **DESNO**, **DOLE** vagy **LEVO**. Ha Pingo már a szélén van (távolság = 0), írja ki: **IVICA**.

#### **D** Kimenet és végső pozíció

Írja ki a kimenetet és Pingo végső pozícióját egy sorban. Formátum:

- ha megszökött: **PINGVIN POBEGAO NA (X,Y)**,
- ha elkapták: **LAV UHVATIO NA (X,Y)**.

#### **E** Válasz mátrix feliratokkal

Írja ki a **10×10**-es mátrixot **sor- és oszlopfeliratokkal** (0–9 számok).

A következő szimbólumokat kell használni:

- **'P'** – Pingo kezdeti pozíciója (**(r1, c1)** mező),
- **'L'** – Leo pozíciója (**(r2, c2)** mező),
- **'p'** – az az út, amelyen Pingo áthaladt (a kezdeti és végső pozíció nélkül),
- **'F'** – végső mező, ha Pingo megszökött (a szélén),
- **'X'** – az a mező, ahol Leo elkapta Pingót (egybeesik **'P'**-vel – ez esetben csak **'X'**-et kell írni),
- **'-'** – összes többi mező.

Kimeneti formátum: minden sor elé írja a sorszámát, a mátrix fölé az oszlopszámok sorát (lásd a teszt példát).

---

★ Sok sikert a versenyen! ★